

日本国特許庁  
JAPAN PATENT OFFICE

#2  
9A  
82101

J1000 U.S. PTO  
09/864300



別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出願年月日

Date of Application:

2000年 5月31日

出願番号

Application Number:

特願2000-161349

出願人

Applicant(s):

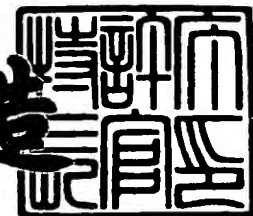
日本電気株式会社



2001年 4月27日

特許庁長官  
Commissioner,  
Japan Patent Office

及川耕造



出証番号 出証特2001-3034935

【書類名】 特許願

【整理番号】 49210429

【提出日】 平成12年 5月31日

【あて先】 特許庁長官殿

【国際特許分類】 H04L 12/28

【発明者】

    【住所又は居所】 東京都港区芝五丁目 7 番 1 号 日本電気株式会社内

    【氏名】 西原 基夫

【特許出願人】

    【識別番号】 000004237

    【氏名又は名称】 日本電気株式会社

【代理人】

    【識別番号】 100088812

    【弁理士】

    【氏名又は名称】 ▲柳▼川 信

【手数料の表示】

    【予納台帳番号】 030982

    【納付金額】 21,000円

【提出物件の目録】

    【物件名】 明細書 1

    【物件名】 図面 1

    【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 パイプライン処理型シェーピング装置およびその方法

【特許請求の範囲】

【請求項 1】 複数フローの入力パケットに関して、パイプライン処理部によりパイプライン処理を行いつつこれ等各フローのシェーピングを行ってスケジューリング予定時刻を算出するようにしたパイプライン処理型シェーピング装置であって、前記フロー毎に前記パイプライン処理部で仕掛かり中のフロー情報を管理格納する格納手段と、前記パイプライン処理部へ入力されたパケットのフローに対応する前記フロー情報を参照して、当該フローに属する全てのパケットをつないだ仮想的なパケットが入力されたものとして前記スケジューリング予定時刻を算出する算出手段とを含むことを特徴とするパイプライン処理型シェーピング装置。

【請求項 2】 前記算出手段は、前記パイプライン処理部へのパケットの入力に応答してこのパケットが属するフローの前記フロー情報を前記格納手段から読出す手段と、この読出し情報を参照して前記スケジューリング予定時刻を算出する手段とを有することを特徴とする請求項 1 記載のパイプライン処理型シェーピング装置。

【請求項 3】 前記パケットの前記パイプライン処理部への入力に応答して、前記格納手段のフロー情報を前記フロー毎に更新する格納情報更新手段を含むことを特徴とする請求項 1 または 2 記載のパイプライン処理型シェーピング装置。

【請求項 4】 前記格納手段は前記パイプライン処理部の処理ブロック数に等しい内部レジスタを有し、前記内部レジスタの各々は、パイプライン処理仕掛かり中の同一フローに属するパケットの前記フロー情報を格納するようにしたことを特徴とする請求項 1 ～ 3 いずれか記載のパイプライン処理型シェーピング装置。

【請求項 5】 前記フロー情報は前記パケット長の総和を含むことを特徴とする請求項 1 ～ 4 いずれか記載のパイプライン処理型シェーピング装置。

【請求項 6】 前記複数フローの入力パケットに関して、パイプライン処理部によりパイプライン処理を行いつつこれ等各フローのシェーピングを行ってスケジューリング予定時刻を算出するようにしたパイプライン処理型シェーピング方法であって、前記フロー毎に前記パイプライン処理部で仕掛かり中のフロー情報を管理格納するステップと、前記パイプライン処理部へ入力されたパケットのフローに対応する前記フロー情報を参照して、当該フローに属する全てのパケットをつないだ仮想的なパケットが入力されたものとして前記スケジューリング予定時刻を算出する算出ステップとを含むことを特徴とするパイプライン処理型シェーピング方法。

【請求項 7】 前記算出ステップは、前記パイプライン処理部へのパケットの入力に応答してこのパケットが属するフローの前記フロー情報を前記格納手段から読出すステップと、この読出し情報を参照して前記スケジューリング予定時刻を算出するステップとを有することを特徴とする請求項 6 記載のパイプライン処理型シェーピング方法。

【請求項 8】 前記パケットの前記パイプライン処理部への入力に応答して、前記格納手段のフロー情報を前記フロー毎に更新するステップを含むことを特徴とする請求項 6 または 7 記載のパイプライン処理型シェーピング方法。

【請求項 9】 前記格納手段は前記パイプライン処理部の処理ブロック数に等しい内部レジスタを有し、前記内部レジスタの各々に、パイプライン処理仕掛かり中の同一フローに属するパケットの前記フロー情報を格納するようにしたことを特徴とする請求項 6 ～ 8 いずれか記載のパイプライン処理型シェーピング方法。

【請求項 10】 前記フロー情報は前記パケット長の総和を含むことを特徴とする請求項 6 ～ 9 いずれか記載のパイプライン処理型シェーピング方法。

【請求項 11】 前記複数フローの入力パケットに関して、パイプライン処理部によりパイプライン処理を行いつつこれ等各フローのシェーピングを行ってスケジューリング予定時刻を算出するようにしたパイプライン処理型シェーピング方法の制御プログラムを記録した記録媒体であって、前記制御プログラムは、前記フロー毎に前記パイプライン処理部で仕掛かり中のフロー情報を管理格納す

るステップと、前記パイプライン処理部へ入力されたパケットのフローに対応する前記フロー情報を参照して、当該フローに属する全てのパケットをつないだ仮想的なパケットが入力されたものとして前記スケジューリング予定時刻を算出する算出ステップとを含むことを特徴とする記録媒体。

【請求項 1 2】 前記算出ステップは、前記パイプライン処理部へのパケットの入力に応答してこのパケットが属するフローの前記フロー情報を前記格納手段から読出すステップと、この読出し情報を参照して前記スケジューリング予定時刻を算出するステップとを有することを特徴とする請求項 1 1 記載の記録媒体。

【請求項 1 3】 前記パケットの前記パイプライン処理部への入力に応答して、前記格納手段のフロー情報を前記フロー毎に更新するステップを含むことを特徴とする請求項 1 1 または 1 2 記載の記録媒体。

【請求項 1 4】 前記格納手段は前記パイプライン処理部の処理ブロック数に等しい内部レジスタを有し、前記内部レジスタの各々に、パイプライン処理仕掛けり中の同一フローに属するパケットの前記フロー情報を格納するようにしたことを特徴とする請求項 1 1 ～ 1 3 いずれか記載の記録媒体。

【請求項 1 5】 前記フロー情報は前記パケット長の総和を含むことを特徴とする請求項 1 1 ～ 1 4 いずれか記載の記録媒体。

【発明の詳細な説明】

【0 0 0 1】

【発明の属する技術分野】

本発明はパイプライン処理型シェーピング装置およびその方法に関し、特にネットワークを構成する基幹通信装置からアクセス通信装置間へのインタフェースにおいて個々のコネクション毎にパケットもしくはセルのスケジューリング予定時刻を、パイプライン処理方式にて決定するためのスケジューリング予定時刻算出方式に関するものである。

【0 0 0 2】

【従来の技術】

昨今のデータトラヒックの増加により、基幹通信装置とアクセス通信装置間の

インタフェースの高速化が進んでいる。アクセス装置と接続する基幹装置のインタフェースにおいては、ユーザとの契約に基づいたコネクション単位に厳密なポリシングやシェーピングを実施する必要がある。このコネクションは、A T M ( Asynchronous Transfer Mode ) インタフェースの場合、V C ( Virtual Connection ) に該当し、P O S ( Packet over SONET ) インタフェースの時は、I P ( Internet Protocol ) フローに該当する。代表的なポリシング処理、シェーピング処理としては、トークンバケット方式があげられる。

## 【 0 0 0 3 】

このトークンバケット方式は、アルゴリズムの処理時間がかかるため、従来の基幹通信装置ではパイプライン処理により実現している。パイプライン処理とは、図 1 0 に示すように、トークンバケット方式のアルゴリズムを複数の処理に分割し、個々の処理は最小パケット長  $T$  に該当する時間で実行するものである。

## 【 0 0 0 4 】

## 【 発明が解決しようとする課題 】

しかしながら、特にシェーピング処理において、同じコネクションに属する最小パケットが連続して入力すると以下の問題がある。すなわち、シェーピングの場合、図 1 0 で処理 1 にあるコネクションのパケット A が入力されて、パケット A の転送予定の時刻が決定するのは処理 N が終了する時である。従って、図 1 1 に示すように、同じコネクションに属する次のパケット B のパイプライン処理を開始するためには、パケット A に関する処理 N の終了の後になる。従って、パケット A とパケット B との間隔が、処理 1 ～ N までの時間  $N \times T$  よりも短い時、個々のパケットの転送予定時刻を実時間で判定することができず、結果としてパケット A とパケット B との間隔が  $N \times T$  以上になるような、比較的ピークレート ( peak-rate ) の遅いコネクションにしか、シェーピング処理を行うことができない。結果として、高速インタフェースにおいては、ピークレートの高いコネクションに対するパケット転送のシェーピングを実現することが困難であった。

## 【 0 0 0 5 】

本発明の目的は、従来のシェーピングに関するパイプライン処理を改善することで、任意の速度のコネクションに対しても厳密なシェーピング処理を、簡易な

回路構成の追加により実現可能としたパイプライン処理型シェーピング装置およびその方法を提供することである。

【0006】

【課題を解決するための手段】

本発明によれば、複数フローの入力パケットに関して、パイプライン処理部によりパイプライン処理を行いつつこれ等各フローのシェーピングを行ってスケジューリング予定時刻を算出するようにしたパイプライン処理型シェーピング装置であって、前記フロー毎に前記パイプライン処理部で仕掛かり中のフロー情報を管理格納する格納手段と、前記パイプライン処理部へ入力されたパケットのフローに対応する前記フロー情報を参照して、当該フローに属する全てのパケットをつないだ仮想的なパケットが入力されたものとして前記スケジューリング予定時刻を算出する算出手段とを含むことを特徴とするパイプライン処理型シェーピング装置が得られる。

【0007】

そして、前記算出手段は、前記パイプライン処理部へのパケットの入力に応答してこのパケットが属するフローの前記フロー情報を前記格納手段から読出す手段と、この読出し情報を参照して前記スケジューリング予定時刻を算出する手段とを有することを特徴とする。また、前記パケットの前記パイプライン処理部への入力に応答して、前記格納手段のフロー情報を前記フロー毎に更新する格納情報更新手段を含むことを特徴とする。更に、前記格納手段は前記パイプライン処理部の処理ブロック数に等しい内部レジスタを有し、前記内部レジスタの各々は、パイプライン処理仕掛かり中の同一フローに属するパケットの前記フロー情報を格納するようにしたことを特徴とし、そして、前記フロー情報は前記パケット長の総和を含むことを特徴とする。

【0008】

本発明によれば、前記複数フローの入力パケットに関して、パイプライン処理部によりパイプライン処理を行いつつこれ等各フローのシェーピングを行ってスケジューリング予定時刻を算出するようにしたパイプライン処理型シェーピング方法であって、前記フロー毎に前記パイプライン処理部で仕掛かり中のフロー情

報を管理格納するステップと、前記パイプライン処理部へ入力されたパケットのフローに対応する前記フロー情報を参照して、当該フローに属する全てのパケットをつないだ仮想的なパケットが入力されたものとして前記スケジューリング予定時刻を算出する算出ステップとを含むことを特徴とするパイプライン処理型シェーピング方法が得られる。

## 【 0 0 0 9 】

そして、前記算出ステップは、前記パイプライン処理部へのパケットの入力に  
応答してこのパケットが属するフローの前記フロー情報を前記格納手段から読出  
すステップと、この読出し情報を参照して前記スケジューリング予定時刻を算出  
するステップとを有することを特徴とする。また、前記パケットの前記パイプ  
ライン処理部への入力に  
応答して、前記格納手段のフロー情報を前記フロー毎に更  
新するステップを含むことを特徴とする。更に、前記格納手段は前記パイプ  
ライン処理部の処理ブロック数に等しい内部レジスタを有し、前記内部レジスタの各  
々に、パイプライン処理仕掛かり中の同一フローに属するパケットの前記フロー  
情報を格納するようにしたことを特徴とし、前記フロー情報は前記パケット長の  
総和を含むことを特徴とする。

## 【 0 0 1 0 】

本発明によれば、前記複数フローの入力パケットに関して、パイプライン処理  
部によりパイプライン処理を行いつつこれ等各フローのシェーピングを行ってス  
ケジューリング予定時刻を算出するようにしたパイプライン処理型シェーピング  
方法の制御プログラムを記録した記録媒体であって、前記制御プログラムは、前  
記フロー毎に前記パイプライン処理部で仕掛かり中のフロー情報を管理格納する  
ステップと、前記パイプライン処理部へ入力されたパケットのフローに対応する  
前記フロー情報を参照して、当該フローに属する全てのパケットをつないだ仮想  
的なパケットが入力されたものとして前記スケジューリング予定時刻を算出する  
算出ステップとを含むことを特徴とする記録媒体が得られる。

## 【 0 0 1 1 】

本発明の作用を述べる。パイプライン処理と連動するキャッシュ部（格納手段）  
を用意し、このキャッシュ部において、パイプライン処理部で仕掛かり中のパ



ケットのフロー情報を管理し、同じフローに属するパケットがある場合は、キャッシュ部が該当のパケットを全てつなぎ合わせた仮想的なパケットを想定したパラメータをパイプライン処理部に渡し、パイプライン処理部では、この仮想的なパラメータを元にパイプライン処理を実行することにより、任意のピークレート（同じフローに属する入力パケット間隔の逆数）の速度を有するフローに対しても、またどのような高速な伝送路インタフェースにおいても、常にシェーピングによるスケジューリング予定時刻をリアルタイムで計算できるという効果が得られる。

## 【 0 0 1 2 】

## 【発明の実施の形態】

以下に図面を参照しつつ本発明の実施例を説明する。図 1 を参照すると、本発明の一実施例としてのスケジューリング機能を行うシェーピング判定部の全体構成が示されている。ここで、図 2 を参照すると、本発明の目的とするシェーピング機能がネットワーク内で配備される位置を示している。図 2 に示すように、ネットワークはユーザ通信装置 A, G と、アクセス通信装置 B, F と、バックボーンである基幹網を構築する基幹通信装置 C, D, E とからなる。アクセス通信装置は、ユーザ通信装置と接続され、ユーザから来たトラヒックを束ねて基幹通信装置に送る。それらのトラヒックは、複数の基幹通信装置を経て対局のアクセス通信装置に到達し、その後ユーザ通信装置に転送される。

## 【 0 0 1 3 】

ここで、トラヒックの種類として、IP フローもしくは ATM セルを想定する。本発明で述べるシェーピング機能は、基幹通信装置からアクセス通信装置に転送するトラヒック、もしくはアクセス通信装置からユーザ通信装置に転送するトラヒックに対して行われる機能である。具体的には、個々の IP フローや ATM コネクションに属するパケットに対して、IP フローや ATM コネクションに規定される平均速度、ピーク速度を守るようにスケジューリングした上で伝送路に転送する機能であり、通常はパケット単位にどの時刻に転送するかを決定する処理である。

## 【 0 0 1 4 】

図 3 はさらに基幹通信装置（例えば、図 2 の C）における本発明のシェーピング機能の適用位置を示す。図 1 で示される本発明の回路は図 3 のシェーピング判定部 1 3 に相当するものである。そもそも、基幹通信装置は、スイッチ 1 0 と I F カード 1 1 とから構成され、I F カードを介して入力してきたパケットを適正な出力ポートにスイッチングし、出力ポートに該当する I F カードではシェーピング機能を含む複数の処理を行って、伝送路にパケット転送する。

## 【 0 0 1 5 】

図 3 はシェーピング機能に関するブロックのみ示している。スイッチから転送されてくるパケットは、図 3 で示す I F カード 1 1 のバッファ部 1 2 にまず蓄積される。バッファ部 1 2 が個々のパケットの制御情報をシェーピング判定部 1 3 にパケット情報として渡す。シェーピング判定部 1 3 は、それらのパケット情報を元に、個々のパケット毎に転送すべきスケジューリング時刻を計算し、内部のスケジューリング予定時刻メモリに格納する。

## 【 0 0 1 6 】

図 3 で示す I F カード 1 1 のパケット読み出し部 1 4 は、常に時計 1 5 の時刻と上記スケジューリング予定時刻メモリとの内容を比較し、もし現在の時刻と同じスケジューリング予定時刻を有するパケットが登録されてあれば、そのパケットをバッファ部 1 2 から読み出し、さらに伝送路に転送するものである。

## 【 0 0 1 7 】

ここで、図 1 のブロック図に戻ると、本発明の実施例は、アクセス通信装置と接続される基幹通信装置で実現されるシェーピング処理において、複数の処理ブロックからなるパイプライン処理部 2 0 と、このパイプライン処理部内の処理ブロック 1 ～ 7 と協調して動作するキャッシュ部 2 2 と、パケットのスケジューリング予定時刻を格納するスケジューリング予定時刻メモリ 2 3 と、個々のパケットの属するフローに関する情報を管理する管理メモリ 2 1 と、パイプライン処理部 2 0 に付随する時計部 2 4 とを有している。

## 【 0 0 1 8 】

さらに、キャッシュ部 2 2 においては、内部レジスタ部 2 6 と、フロー検索部 2 5 と、レジスタ更新部 2 7 とが設けられている。内部レジスタ部 2 6 には、パ

パイプライン処理部 2 0 における処理ブロック 1 ～ 7 の数と同じ数の内部レジスタ（分かりやすくするために、各レジスタも 1 ～ 7 として示している）が設けられている。

#### 【 0 0 1 9 】

尚、図 4 は図 1 のブロック図における各ブロック間の信号内容①～⑦を説明するものであり、また図 5（A）は管理メモリ 2 1 の内容を、（B）はキャッシュ部 2 2 の内部レジスタの内容を、また（C）はスケジューリング予定時刻メモリ 2 3 の内容を夫々示している。これ等に図 4，5 ついては、後述する。

#### 【 0 0 2 0 】

パイプライン処理部 2 0 において、処理ブロック 1 ～ 7 が図 6 で示されるアルゴリズムを実行する。またパイプライン処理に付随して、図 1 のキャッシュ部 2 が図 6 の処理アルゴリズムに基づき動作する。

#### 【 0 0 2 1 】

図 1 に示すように、キャッシュ部 2 2 は内部レジスタ部 2 6、フロー検索部 2 5、レジスタ更新部 2 7 から構成され、内部レジスタ部 2 6 はパイプライン処理部 2 0 の処理ブロック 1 ～ 7 において処理中のパケットに関するフロー情報を保持する。

#### 【 0 0 2 2 】

管理メモリ 2 1 の構成は図 5（A）で示すように、フロー識別子をアドレスとして、「トークン加算値」、「トークン加算間隔」、「トークン値」、「最新スケジューリング時刻」の情報を有する。また、キャッシュ部 2 2 の内部レジスタは図 5（B）で示すように、レジスタが空きであるか否かを示す「有効ビット（valid-bit）」、「フロー識別子」、パイプライン処理部内においてフローに属するパケットの数を示す「パケット数」、当該パケットの長さの総和である「パケット長総和」からなる。なお「パケット数」は、パイプライン処理部 2 0 内に存在する（パイプライン処理仕掛かり中）フローに対してのみ規定される値であり、キャッシュ部 2 2 がすべてのフローを管理する必要はない。

#### 【 0 0 2 3 】

パイプライン処理部 2 0 の各処理ブロック 1 ～ 7 は、入力パケットに関して、

パイプライン的に図 6 の処理アルゴリズムで示される演算を行う。各処理ブロック 1 ～ 7 は、自分より後段の処理ブロックにおいて、自分が処理中のパケットと同じフローに属するパケットが存在しない場合、管理メモリから得た同フローの前のパケットの最新スケジューリング時刻を元に、次に転送できるスケジューリング予定時刻を計算する。自分より後段の処理ブロックにおいて、自分が処理中のパケットと同じフローに属するパケットが存在する場合、それらのパケットと自分のパケットをつなぎ合わせた仮想的なパケットを想定し、左記パケットが転送できるスケジューリング時刻を計算する。

## 【 0 0 2 4 】

このように、パイプライン処理部のみで構成される従来のシェーピング方式に比べ、パイプライン処理と連動するキャッシュ部を用意し、キャッシュ部においてパイプライン処理部で仕掛かりのパケットのフロー情報を管理し、同じフローに属するパケットがある場合は、キャッシュ部が該当のパケットを全てつなぎ合わせた仮想的なパケットを想定したパラメータをパイプライン処理部に渡し、パイプライン処理部では左記の仮想的なパラメータを元にパイプライン処理を実行することにより、任意のピークレート（同じフローに属する入力パケット間隔の逆数）の速度を有するフローに対しても、またどのような高速な伝送路インタフェースにおいても、常にシェーピングによるスケジューリング予定時刻をリアルタイムで計算できる効果が得られる。

## 【 0 0 2 5 】

図 3 のシェーピング判定部 1 3 には、バッファ部 1 2 からパケット情報が転送されてくる。このパケット情報は図 1 のパイプライン処理部 2 0 の処理ブロック 1 に入力される。処理ブロック 1 はパケット情報内のフロー識別子を元に管理メモリ 2 1 にアクセスし、該当のパケットのフロー情報であって、パイプライン演算処理のための演算パラメータ（トークン加算値  $T_K$ 、トークン加算間隔  $L$ 、トークン値  $P$ 、最新スケジューリング時刻  $R_T$ ）を得る。また処理ブロック 1 はフロー識別子をキャッシュ部 2 2 に通知する。

## 【 0 0 2 6 】

キャッシュ部 2 2 は、図 6 で示すように、フロー識別子を元に、内部のフロー

検索部 2 5 と内部レジスタ部 2 6 の動作により、該当のフロー識別子に関する情報が内部レジスタに登録されてあるか否かを検索し、登録されている場合はその登録情報であるパケット数、パケット長総和などの情報を処理ブロック 1 に通知する。

## 【 0 0 2 7 】

処理ブロック 2 は処理ブロック 1 から得るフロー情報とキャッシュ部 2 2 から得るフロー情報を元に、図 6 に規定される処理を行う。またそれに伴い、再びキャッシュ部にフロー識別子と自分のパケット長  $b$  とを送る。同時に、処理ブロック 2 で演算した結果を処理ブロック 3 に転送する。

## 【 0 0 2 8 】

処理ブロック 3 は処理ブロック 2 から転送される情報を元に図 6 で示す処理アルゴリズムを実行し、結果を処理ブロック 4 に転送する。処理ブロック 4 は処理ブロック 3 から転送される情報を元に図 6 で示す処理アルゴリズムを実行し、結果を処理ブロック 5 に転送する。

## 【 0 0 2 9 】

処理ブロック 5 は処理ブロック 4 から転送される情報と時計部から転送される現在時刻  $N T$  とを元に、図 6 で示す処理アルゴリズムを実行し、結果を処理ブロック 6 に転送する。処理ブロック 6 は処理ブロック 5 から転送される情報を元に図 6 で示す処理アルゴリズムを実行し、結果を処理ブロック 7 に転送する。処理ブロック 7 は処理ブロック 6 から転送される情報を元に、図 6 で示す処理アルゴリズムを実行し、結果を管理メモリ 2 1、スケジューリング予定時刻メモリ 2 3、キャッシュ部 2 2 に転送する。

## 【 0 0 3 0 】

具体的には、管理メモリ 2 1 に対しては、フロー識別子をアドレスとしてアクセスし、トークン加算値、トークン加算間隔、トークン値、最新スケジューリング時刻を更新する。また、スケジューリング予定時刻メモリ 2 3 に対しては、スケジューリング予定時刻とパケット識別子に登録する。またキャッシュ部 2 2 に対しては、内部レジスタの内容を変更するために、フロー識別子、パケット長  $b$  を送る。時計部 2 4 は現在時刻  $N T$  を常に処理ブロック 5 に通知する。尚、図 7

は図 6 に示したアルゴリズムに使用される変数（パラメータ）を説明する図である。

#### 【 0 0 3 1 】

キャッシュ部 2 2 は処理ブロック 2 からフロー識別子、パケット長 b を受け、また処理ブロック 7 からフロー識別子、パケット長 b を受け、図 6 で示す処理アルゴリズムに基づき、フロー検索部 2 5、レジスタ更新部 2 7、内部レジスタ部 2 6 の相互処理により、内部レジスタの更新を行う。上記にあげた処理のタイミングとして、パイプライン中にフロー 2 に属する複数のパケットが無い場合の処理タイミングのアクセスを図 8 に示し、パイプライン中にフロー 2 に属する複数のパケットが有る場合の処理タイミングのアクセスを図 9 に示している。

#### 【 0 0 3 2 】

以下、本実施例の動作につき説明する。まず、キャッシュ付きパイプライン型シェーピング回路の適用される箇所について説明する。図 2 で示すように、ネットワークはユーザ通信装置 A、G、アクセス通信装置 B、F および基幹通信装置 C、D、E により構成され、ユーザ装置 A、G 間の通信はアクセス通信装置 B、基幹通信装置 C、基幹通信装置 D、基幹通信装置 E、アクセス通信装置 F を介して行われる。特に、基幹通信装置→アクセス通信装置もしくはアクセス通信装置→ユーザ通信装置の箇所においては、ユーザ装置間におけるフロー単位にシェーピングの処理を行う。このシェーピング機能は、図 2 において、斜線で示される。

#### 【 0 0 3 3 】

次に、基幹通信装置 C の概要を図 3 に示す。基幹通信装置は複数の I F カード 1 1 とスイッチ 1 0 とから構成される。伝送路から I F カードに入力されたパケットもしくはセルは、スイッチ 1 0 にて交換処理された後に、出力すべき伝送路と接続する I F カードに転送される。スイッチ 1 3 から伝送路の送信方向におけるシェーピング機能の処理は、図 3 に示す通り、I F カード内のバッファ部 1 2、シェーピング判定部 1 3、パケット読み出し制御部 1 4、時計 1 5 の組合せにより実現される。

#### 【 0 0 3 4 】

スイッチ 1 0 から転送されてきたパケットもしくはセルは、まずバッファ部 1 2 に蓄積される。蓄積されると同時に、バッファ部 1 2 は該当パケットに関する情報をパケット情報としてシェーピング判定部 1 3 に通知する。シェーピング判定部 1 3 は、ユーザ間通信を特定するフロー識別子を元に、個々のパケットの伝送路へのスケジューリング予定時刻を計算し、内部のスケジューリング予定時刻メモリに登録する。

## 【 0 0 3 5 】

一方、パケット読み出し制御部 1 4 は時計部 1 5 から現在時刻の通知を受け、現在時刻に転送するはずのパケットがバッファ部 1 2 に滞留しているか否かを、シェーピング判定部 1 3 内のスケジューリング予定時刻メモリに問い合わせ、もし該当のパケットが登録されてあれば、そのパケットをバッファ部から読み出し、さらに伝送路上に転送する。以上の動作により、個々のユーザ通信装置間におけるフロー（フロー識別子で特定される）毎に、シェーピング機能を実現することができる。

## 【 0 0 3 6 】

なお、スケジューリング予定時刻メモリ 2 3 において、同時刻に複数のフローのパケットが登録されている場合、パケット読み出し制御部 1 4 はこれ等複数のパケットの中から、一つのパケットのみを選択して読み出しを行う必要があるが、本発明は、この様な機能の実現に関するものではなく、シェーピング判定部 1 3 においてスケジューリング予定時刻を決定し、スケジューリング予定時刻メモリ 2 3 に登録する処理に関するものであるので、その部分のみを説明する。特に、OC-48 (2.4 Gbps)、OC-192 (10 Gbps) 以上の高速回線を終端する IF カードにおいて、本発明は任意のシェーピングパラメータ（ピーク速度、平均速度等）に対して正確なスケジューリング予定時刻を決定することができる。

## 【 0 0 3 7 】

以下、本発明における実施例につき説明する。図 1 で示すように、図 3 のシェーピング判定部 1 3 はパイプライン処理部 2 0、管理メモリ 2 1、キャッシュ部 2 2、スケジューリング予定時刻メモリ 2 3、時計部 2 4 から構成される。図 3

のバッファ部 1 2 からパケット情報がパイプライン処理部に転送される。パケット情報の内容は、フロー識別子、パケット識別子、パケット長  $b$  の 3 つである。フロー識別子は、該当のパケットが属するユーザ装置間におけるフローを示す識別子であり、シェーピング判定部 1 3 の入力において予め付与されている。パケット識別子は、図 3 のバッファ部 1 2 内に滞留するパケットを特定する識別子であり、シェーピング判定部 1 3 の入力において予め付与されている。パケット長  $b$  は該当パケットの長さを示す。

## 【 0 0 3 8 】

パイプライン処理部 2 0 は図 1 で示すように、処理ブロック 1 ～ 7 より構成される。処理ブロック 1 ～ 7 には一定の処理が規定されており、各処理ブロックは前段の処理ブロックから処理結果を受け取り、自分に割り当てられている規定の処理を行った後、次の処理ブロックに結果を転送する。各ブロックの処理は、図 6 に記載されている。図 8, 9 は処理ブロックのタイミングを示し、処理 1 ～ 7 がそれぞれ処理ブロック 1 ～ 7 の処理タイミングを示す。処理 1 ～ 7 が実施された後、パケットのスケジューリング予定時刻が決定される。

## 【 0 0 3 9 】

処理ブロック間で転送される情報は、図 1 の①に示す通り、パケット情報からのフロー識別子、パケット識別子、パケット長  $b$  と、管理メモリ 2 1 から読み出されるトークン加算値  $TK$ 、トークン加算間隔  $L$ 、トークン値  $P$ 、最新スケジューリング時刻  $RT$  と、内部変数  $fbit$ 、 $X$ 、 $Y$ 、 $Z$ 、 $W$ 、 $f2bit$ 、 $Y2$  からなる。これらの情報は、図 7 に示されており、個々の処理ブロックが自律的に処理を行う上で必要なパラメータであり、個々の入力パケット毎に、処理ブロックを介して処理ブロック 1 から処理ブロック 7 に転送される。

## 【 0 0 4 0 】

管理メモリ 2 1 には、アドレスとして、処理ブロック 1 もしくは処理ブロック 7 からフロー識別子が与えられる。データとして、システム立ち上げ時に規定されるシェーピングパラメータであるトークン加算値  $TK$ 、トークン加算間隔  $L$  がある。さらに、動的に変更する値として、トークン値  $P$ 、最新スケジューリング時刻  $RT$  がある。管理メモリの内容は処理ブロック 1 にて読み出され、処理プロ



ック 7 にて P と R T のみ更新される（図 1 の①と⑤のアクセス）。

【 0 0 4 1 】

スケジューリング予定時刻は連想メモリ（CAM: Content Addressable Memory）からなり、パケット識別子とそのスケジューリング予定時刻が登録されている。図 3 のパケット読み出し制御部 1 4 は、現在時刻と一致するスケジューリング予定時刻の登録があるか否か、スケジューリング予定時刻メモリ 2 3 にて検索し、登録がある場合にそのパケット識別子を元に、バッファ部 1 2 からパケットを読み出す。

【 0 0 4 2 】

キャッシュ部 2 2 はフロー検索部 2 5、レジスタ更新部 2 7、内部レジスタ部 2 6 から構成される。キャッシュ部 2 2 はパイプライン処理部 2 0 の各処理ブロックで処理仕掛かり中のパケットのフロー情報を管理する。具体的には、各処理ブロックで処理中のパケットのフロー情報が、内部レジスタ部 2 6 の個々の内部レジスタに格納されている。パイプライン処理部内で処理中のフローの数は、個々の処理ブロックがすべて異なるフローのパケットを処理している時、最大になる。従って、内部レジスタ部の内部レジスタとして、処理ブロックの数だけ用意すれば良い。

【 0 0 4 3 】

内部レジスタの構成は、図 5（B）に示したように、本レジスタが有効であるか否かを示す有効ビット、フローを特定するフロー識別子、左記フローに属するパケットの数 K（パイプライン処理部で処理仕掛かり中のパケットの中において）、左記フローに属するパケットのパケット長の総和 B となる。

【 0 0 4 4 】

時計部 2 4 は、現在時刻を管理し、処理ブロック 5 に常時、現在時刻 N T を通知する。なお本時計部の時刻 N T は、図 3 の I F カード 1 1 上の時計の時刻より、パイプライン処理の遅延に依存する時間  $\tau$  だけ進んでいるものとする。個々の処理ブロックの処理の流れは図 6 に示す通りである。

【 0 0 4 5 】

以下に、詳細を説明する。まず、パケット情報はパイプライン処理部の処理ブ

ロック 1 に入力される。処理ブロック 1 はパケット情報内のフロー識別子を元に管理メモリ 2 1 にアクセスし、該当のパケットのフロー情報（トークン加算値  $T_K$ 、トークン加算間隔  $L$ 、トークン値  $P$ 、最新スケジューリング時刻  $R_T$ ）を得る（図 1 の②のアクセス）。処理ブロック 1 はさらにフロー識別子をキャッシュ部 2 2 に通知する。キャッシュ部 2 2 のフロー検索部 2 5 は図 6 で示すように左記フロー識別子を元に、内部レジスタ部 2 6 において、該当のフロー識別子に関する情報が内部レジスタに登録されてあるか否かを検索し、登録されている場合は、登録情報であるパケット数  $K$ 、パケット長総和  $B$  を処理ブロック 1 に通知する（図 1 の③のアクセス）。

## 【 0 0 4 6 】

処理ブロック 2 は処理ブロック 1 およびキャッシュ部 2 2 からパケットの属するフローの情報を受け取り、処理ブロック 2 ～ 7 において、自分の有するパケットと同じフローに属するパケットがあるか否かで異なる処理を行う。自分の有するパケットと同じフローに属するパケットがない場合は、処理ブロック 1 にて管理メモリ 2 1 から読み出された情報のみを元にしてシェーピングのスケジューリング予定時刻判定を行う。

## 【 0 0 4 7 】

自分の有するパケットと同じフローに属するパケットがある場合は、同様にシェーピングのスケジューリング予定時刻判定を行うことができない。というのは、管理メモリ 2 1 にはパイプライン処理部 2 0 で処理中のフローに関する結果が反映されていないからである。そこで、後段の処理ブロック 3 ～ 7 に含まれる同じフローのパケット情報も含めた上で、スケジューリング予定時刻を決定する。具体的には、現状のトークン値  $P$ 、最新スケジューリング時間  $R_T$ 、パケット長 = 「処理ブロック 2 で処理中のパケット長  $b$ 」 + 「処理ブロック 3 ～ 7 に含まれる同じフローのパケット長の総和  $B$ 」を元にスケジューリング予定時刻を決定する。

## 【 0 0 4 8 】

つまり、仮想的にパケット長  $b$  ではなく、パケット長 =  $b + B$  の大きなパケットが入力したものとして、スケジューリング予定時刻の決定を行う。これにより

、パイプライン処理部内に同じフローに属するパケットがある場合でも、常に正しいスケジューリング予定時刻を決定することができる。

【 0 0 4 9 】

また、I Fカード 1 1 の伝送路速度が上がり図 8, 9 のパケット情報転送時間  $t$  が短くなる場合には、さらに細かい処理ブロック 1 ~  $N$  ( $N > 8$ ) に分割することにより、個々の処理ブロックの処理時間  $t$  を短くすることができる。結果として、インタフェースの高速化に関わらずスケジューリング予定時刻を正しく判定することができる。

【 0 0 5 0 】

処理ブロック 2 は、キャッシュ部から得るパケット数  $K$  が 0 の時、処理ブロック 2 のパケットと同一のフローに属するパケットが、後段の処理ブロックに存在しないことを認識する。従って、処理ブロック 1 にて管理メモリから得た最新スケジューリング予定時刻  $R T$  の次に転送できる時刻を決定すれば良い。

【 0 0 5 1 】

まず、現状のトークン値がパケット転送を行うのに足りているか否かの判定を行う。最新のトークン値は管理メモリから得られる  $P$  であり、処理中のパケットの長さは  $b$  なので、 $X = P - b$  の計算を行い、 $X > 0$  ならばトークン値が足りているので、 $R T$  の次の時間にパケットを転送することができ、 $X \leq 0$  ならば、トークンがたまって  $X > 0$  となるまでスケジューリング予定時刻を遅らせる。また  $X > 0$  ならば  $R T$  の次の時間に即時にパケット転送できることを後段の処理ブロックに通知するために、 $fbit = 1$  とし、 $X \leq 0$  の時は  $fbit = 0$  とする。

【 0 0 5 2 】

また、キャッシュ部から得るパケット数  $K > 0$  の時は、後段の処理ブロックに同一のフロー識別子を有するパケットが自分以外に  $K$  個存在するとして認識する。 $K$  個の転送予定時刻はパイプライン処理が終了していないので、決定していない。従って、管理メモリの最新スケジューリング予定時刻  $R T$  は、同じフロー識別子を有する直前のパケットのスケジューリング予定時刻ではなく、 $K$  個前のパケットのスケジューリング予定時刻である。そこで、 $K$  個前のパケットのスケジューリング予定時刻から始まって、 $K + 1$  個目のパケットを転送する時刻を決定

すれば良い。

【 0 0 5 3 】

まず、必要なトークン値を確認するために、 $X = P - b$ の代わりに $X = P - (B + b)$ の計算を行う。 $X$ は該当の packets を転送するために足りないトークン値を示すので、 $X > 0$ ならばトークン値が足りており、 $RT$ の次の時間に packets を転送することができる。もし $X \leq 0$ ならば、トークンがたまって $X > 0$ となるまで、スケジューリング予定時刻を遅らせなければならない。 $X > 0$ ならば $RT$ の次の時間に即時に packets 転送できることを後段の処理ブロックに通知するために、 $fbit = 1$ とする。 $X \leq 0$ の時は、 $fbit = 0$ とする。

【 0 0 5 4 】

処理ブロック 2 が処理している packets により、パイプライン処理部内のフローの情報が変わるので、キャッシュ部が内部レジスタの変更を行わなければならない。そこで、処理ブロック 2 は処理中の packets のフロー識別子、 packets 長さ  $b$  をキャッシュ部に転送する。

【 0 0 5 5 】

処理ブロック 3 は処理ブロック 2 より通知される  $fbit$  および  $X$  を元に packets のスケジューリング予定時刻と、このスケジューリング予定時刻における新たなトークン値を決定するために、まず何回トークン  $TK$  を足せば  $X > 0$  となるかの計算を行う。この加算され得る回数を  $Y$  とする。

【 0 0 5 6 】

処理ブロック 3 は、 $fbit = 0$  の時  $X \leq 0$  なので、 $X > 0$  になるまでに何回トークン量  $TK$  の加算が必要か知るために、 $Y = (|X| + 1) \div TK$  の計算を行う ( $|X|$  は  $X$  の絶対値を示す)。また  $fbit = 1$  の時、既にスケジューリング予定時刻は  $RT$  の次の時間  $RT + 1$  であることが決定しているので、 $Y = (RT + 1 - RT) \div TK = 1 \div TK$  の計算を行う。

【 0 0 5 7 】

上記の  $Y$  は該当のフロー識別子の packets に関し、 $X < 0$  の時は最新スケジューリング予定時刻  $RT$  に  $Y$  回のトークン加算間隔時間  $L$  を経た時刻にトークン値  $P$  が 0 以上になることを示している。

## 【 0 0 5 8 】

処理ブロック 4 は Y を元に該当のパケットのスケジューリング予定時刻とその時のトークン値を決定する。スケジューリング予定時刻は Y の定義より  $Z = Y \times L + R T$  より得られる。また、トークン値は、 $fbit = 1$  の時 ( $X > 0$  の時と同義)、 $W = |X| + 1$  となり ( $W = Y \times T K + X$  と同義)、 $fbit = 0$  の時 ( $X \leq 0$  と同義)、 $W = 1$  となる ( $W = Y \times T K + X$  と同義)。

## 【 0 0 5 9 】

処理ブロック 5 は、処理ブロック 4 で得られたスケジューリング予定時刻 Z とトークン値 W の補正を行う。また処理ブロック 5 は時計部から現在時刻 N T を得る。もし Z が現在の時刻 N T より後の時間の場合は、Z と W の補正の必要はない。後段の処理ブロックに補正処理の必要がないことを通知するために、 $f2bit = 0$  とする。

## 【 0 0 6 0 】

しかしながら、 $N T > Z$  の時、スケジューリング予定時刻は既に過去の時刻であるので、スケジューリング時間 Z は N T に補正し、合わせて W も補正する必要がある。そこで、新たに  $Y 2 = (N T - Z) \div T K$  の演算を行う。さらに、後段の処理ブロックに補正処理の必要を通知するために、 $f2bit = 1$  とする。 $N T \leq Z$  の時は  $f2bit = 0$  とする。

## 【 0 0 6 1 】

処理ブロック 6 は、 $f2bit = 1$  の時に、処理ブロック 5 に引き続いて補正処理を進める。新たなスケジューリング予定時刻として、 $Z = N T$  とする。またトークン値として、 $W = W + N T - Z$  とする。 $F2bi = 0$  の時は何も行わない。

## 【 0 0 6 2 】

処理ブロック 7 は処理ブロック 6 で得られた結果を元に各テーブルの更新を行う。まず、Z とパケット識別子をスケジューリング予定時刻メモリに登録する。またフロー識別子をアドレスとして、管理メモリにアクセスし、新たに T K、L、W、Z の値を書き込む。

## 【 0 0 6 3 】

なお、図 8、9 で示すように、管理メモリの読み出しフェース (R) ・書き込

みフェーズ（W）が重ならないように、各処理時間の前半と後半とで夫々のタイミング規定をすることにより、管理メモリにアクセスする処理ブロック 1 と処理ブロック 7 とがアクセス競合を引き起こすことはない。また、処理ブロック 7 にて処理中のパケットはパイプライン処理部から読み出されるので、キャッシュ部の内部レジスタの情報を更新する必要がある。そこで、フロー識別子、パケット長 b をキャッシュ部に通知する。

## 【 0 0 6 4 】

キャッシュ部はパイプライン処理部内の処理ブロックと同期してパイプライン処理部内で処理中のパケットの属するフローに関する管理を行う。キャッシュ部は内部レジスタ部とフロー検索部と内部レジスタ更新部から構成される。内部レジスタ部には、複数の内部レジスタが存在し、この内部レジスタの数はパイプライン処理部における処理ブロックの数と同じである。

## 【 0 0 6 5 】

内部レジスタには、パイプライン処理部内の処理ブロックで処理中のパケットに関するフローの情報のみ格納されている。内部レジスタは、図 5（B）に示すように、4 つのフィールドから構成される。第 1 に有効ビットである。有効ビットは内部レジスタが使用中か否かを示し、有効ビットが on の時、あるフロー識別子が登録されており、有効ビットが off の時は何も登録されておらず、空き（empty）であることを示す。

## 【 0 0 6 6 】

第 2 にフロー識別子フィールドである。フロー識別子フィールドには、パケットに付随して転送されるフロー識別子が登録される。第 3 にパケット数 K である。パケット数 K は、第 2 のフロー識別子に属するパケットがパイプライン処理部に何個存在するかを示す。第 3 にパケット長の総和 B である。パケット長の総和 B はパイプライン処理部内に存在するフロー識別子のフローのすべてのパケットの長さの総和である。

## 【 0 0 6 7 】

フロー検索部 2 5 は内部レジスタの中で特定のフロー識別子に関するレジスタを探索する。内部レジスタ更新部 2 7 は個々の内部レジスタの更新を行う。また

、各レジスタ内に保持されるパケット数フィールドに対する演算 ( $K = K + 1$  の加算、 $K = K - 1$  の減算)、パケット長総和に対する演算 ( $B = B + b$ 、 $B = B - b$ ) 等を行う。

## 【 0 0 6 8 】

まず、キャッシュ部が処理ブロック 1 からフロー識別子を受け取ると、フロー検索部 2 5 が内部レジスタ部 2 6 の内部レジスタにおいて、フロー識別子と一致し、かつ有効ビットが  $o_n$  となるエントリの検索を行う。一致するエントリが検索できた場合は、該当エントリの内部レジスタの内容 (パケット数  $K$ 、パケット長の総和  $B$ ) を読み出し、パケット数  $K$ 、パケット長の総和  $B$  を処理ブロック 2 に返す。一致するエントリが見つからない場合は、パケット数  $K = 0$  として処理ブロック 2 に返す。

## 【 0 0 6 9 】

またキャッシュ部 2 2 は処理ブロック 2 からフロー識別子、パケット長  $b$  を受信し、左記情報に基づくパケット情報の追加処理を行う。また、処理ブロック 7 からフロー識別子、パケット長  $b$  を受信し、この情報に基づくパケット情報の削除処理を行う。処理ブロック 2 からのフロー識別子を  $flowinfo1$ 、パケット長を  $pktlen1$  と規定し、処理ブロック 7 からのフロー識別子を  $flowinfo2$ 、パケット長を  $pktlen2$  と規定すると、以下の規則に元づいて行われる。

## 【 0 0 7 0 】

$Flowinfo1 = flowinfo2$  の時、内部レジスタ部には既に  $flowinfo1$  のフロー情報が存在しており、処理ブロック 2 からの通知によるパケット追加と処理ブロック 7 からのパケット削除により、内部レジスタのパケット数  $K$  は変わらない。パケット長の総和  $B$  は、処理ブロック 2 からのパケット長追加と処理ブロック 7 からのパケット長削除を反映させる必要がある。

## 【 0 0 7 1 】

まず、フロー検索部は、内部レジスタ部の内部レジスタにおいて、 $flowinfo1$  のフロー識別子と一致し、かつ有効ビットが  $o_n$  となるエントリの検索を行う。このエントリに対して、パケット長の総和  $B = B + pktlen1 - pktlen2$  の計算を行い、該当エントリのレジスタ内容を更新する。

## 【 0 0 7 2 】

Flowinfo1 ≠ flowinfo2 の時、処理ブロック 2 からの通知によるフロー情報の追加処理と、処理ブロック 7 からの通知によるフロー情報の削除処理を同時に行う。まず処理ブロック 2 からの通知によるフロー情報の追加処理について説明する。フロー検索部は、flowinfo1 とフロー識別子が一致し、かつ有効ビットが on となるエントリを検索する。そして、レジスタ更新部は、このエントリに対して、パケット数  $K = K + 1$ 、パケット長の総和  $B = B + \text{pktlen1}$  の演算を行い、レジスタ内容の変更を行う。

## 【 0 0 7 3 】

もし、flowinfo1 とフロー識別子が一致し、かつ有効ビットが on となるエントリが無い場合は、flowinfo1、pktlen1 のフロー情報を新たに登録するために、有効ビットが off となる空きエントリを検索する。このエントリに、フロー識別子、パケット数  $K = 1$ 、パケット長の総和  $B = \text{pktlen1}$  を登録する。

## 【 0 0 7 4 】

Flowinfo1 ≠ flowinfo2 の時、処理ブロック 2 からの通知によるフロー情報の追加処理と、処理ブロック 7 からの通知によるフロー情報の削除処理を同時に行う。まず処理ブロック 2 からの通知によるフロー情報の追加処理について説明する。

## 【 0 0 7 5 】

フロー検索部 2 2 は、flowinfo1 とフロー識別子が一致しかつ有効ビット on となるエントリを検索する。そして、レジスタ更新部は、このエントリに対して、パケット数  $K = K + 1$ 、パケット長の総和  $B = B + \text{pktlen1}$  の演算を行い、レジスタ内容の変更を行う。もし、flowinfo1 とフロー識別子が一致し、かつ有効ビット on となるエントリが無い場合は、flowinfo1、pktlen1 のフロー情報を新たに登録するために、有効ビット off となる空きエントリを検索する。このエントリに、フロー識別子、パケット数  $K = 1$ 、パケット長の総和  $B = \text{pktlen1}$  を登録する。

## 【 0 0 7 6 】

次に、処理ブロック 7 からの通知によるフロー情報の削除処理について説明す



る。フロー検索部は、flowinfo2 とフロー識別子が一致しかつ有効ビット on となるエントリを検索する。そして、レジスタ更新部は、このエントリに対して、パケット数  $K = K - 1$ 、パケット長の総和  $B = B - \text{pktlen2}$  の演算を行い、レジスタ内容の変更を行う。ただし、元々のパケット数  $K = 1$  の時は、パイプライン処理部から flowinfo2 のフローに属するパケットが無くなることを意味しているので、単に有効ビットを off とすれば良い。処理ブロック 7 からの通知による処理は、既に内部レジスタに存在するフロー情報の更新であるため、フロー検索部の処理でエントリが無いケースを想定する必要はない。

## 【 0 0 7 7 】

以上の処理は、限られたフロー数（最大でもパイプライン処理部内の処理ブロックの数）の内部レジスタに関する登録、更新処理であり、キャッシュ部全体を組合せ回路等により最適化して設計することにより、実時間で簡易に実現することが可能である。以上の処理ブロック、キャッシュ部の処理のタイミングを図 8 , 9 に示す。

## 【 0 0 7 8 】

図 8 は、パイプライン中に同一のフローに属するパケットが 1 つしかない場合の処理タイミングである。パケット情報として、フロー 1、フロー 2、フロー 3、フロー 4、フロー 5、フロー 6、フロー 7 が連続しているものの、パイプラインの処理段数は処理 1 ～ 処理 7 の 7 段しかないため、個々の処理ブロックはそれぞれ異なるフローのパケットを処理することになる。図 8 で示すように、管理メモリに関しては、読み出しフェーズ（1）と書き込みフェーズ（4）が処理周期の前半・後半に明確に分離されているため、アクセス競合は発生しない。

## 【 0 0 7 9 】

同様に、スケジューリング予定時刻メモリも処理周期の前半（6）でのみ規定されているのでアクセス競合は発生しない。内部レジスタの場合、読み出しフェーズ（処理周期の前半（2））と書き込みフェーズ（処理周期の後半（3）または（5））は明確に分離されている。しかし、書き込みフェーズにおいて、処理ブロック 2 と処理ブロック 7 とがそれぞれ同時に書き込みのアクセスを行うため、アクセス競合が発生する。この場合でも、既に述べたように内部レジスタの数

はパイプライン処理部内の処理ブロック数（図中では 7 個）に限られているので、内部レジスタのアクセス処理を組合せ回路等で最適化して構成することができ、通常のメモリと異なり処理可能となる。

#### 【 0 0 8 0 】

なお、図 8 はフロー 2 パケットの入力に関するアクセスのみ記載しており、実際には個々のフローのパケットに対して、処理ブロック 1 ～処理ブロック 7 が常に動作しているため、毎処理周期で常に管理メモリ、内部レジスタ、スケジューリング予定時刻メモリへのアクセスが発生している。

#### 【 0 0 8 1 】

図 9 は、図 8 の例とは異なってパイプライン中に同一のフローに属するパケットが複数ある場合の処理タイミングの例である。図 9 では、斜線で示した 2 個のフロー 2 パケットが同時にパイプライン中に存在する。図 9 の（5）で示すように、特に内部レジスタの書き込み処理にて、フロー 2 に関して、処理ブロック 2 と処理ブロック 7 とから同時に更新するアクセスが発生している。この処理に関しても、既に述べたように、処理ブロック 2 からのフロー識別子（flowinfo1）と処理ブロック 7 からのフロー識別子（flowinfo2）が一致した場合に、それぞれのパケット長  $b$ （処理ブロック 2 の場合  $pktlen1$ 、処理ブロック 7 の場合  $pktlen2$ ）を元にした書き込みルールを規定することにより、両者の要求を満足した内部レジスタの更新が可能である。

#### 【 0 0 8 2 】

個々の処理ブロックの機能は、パイプライン処理部に入力するパケットの速度、パイプライン処理部におけるアルゴリズムの困難さに応じて決定される。また実施例では処理ブロック 1 ～ 7 で行っているが、必要に応じて処理ブロック数を増やし、処理ブロック 1 ～  $N$ （ $N \geq 8$ ）としても良い。パイプライン処理部に入力するパケットの速度が速いほど、またパイプライン処理部におけるアルゴリズムが困難であるほど、処理ブロックの数を増やす必要がある。また、内部レジスタ部における内部レジスタの数は処理ブロックの数と一致するので、内部レジスタの数も処理ブロックの数と同様に増える。

#### 【 0 0 8 3 】

上記実施例では、内部レジスタ部の構成として、複数の内部レジスタが独立に存在し、フロー識別子を元に内部レジスタを検索したり更新できる機能を規定している。内部レジスタ部の左記機能をすべて連想メモリを使用して実現しても良い。上記実施例では、シェーピング処理に基づいて記述している。しかしながら、本発明の特徴は、上述したように、イプライン処理部に付随してキャッシュ部を設け、キャッシュ部内にはパイプライン処理部で仕掛かり中のパケットのフロー情報を管理し、パイプライン処理部ではキャッシュ部のフロー情報に基づき、仮想的なスケジューリングパラメータで処理を行うという点にある。つまり、シェーピング処理のアルゴリズムに依存せずに、さまざまな複雑なアルゴリズムに適応できる。例えば、シェーピング処理ではなく、パケット廃棄に関するトークンパケットベースのポリシング処理も、同様に実現可能である。

【 0 0 8 4 】

【発明の効果】

以上述べたように、パケットのフロー単位にシェーピングを行う技術に関して、パケット単位に処理が移動するパイプライン処理部に付随して連動するキャッシュ部を設け、このキャッシュ部においてパイプライン処理部内で処理中のパケットのフロー情報を管理する内部レジスタを有し、個々のパケットのスケジューリング時間の計算において、キャッシュ部からの通知により同一のフローに属するパケットが存在しない時は従来方法と同様に、同フローに属する 1 個前のパケットのスケジューリング予定時刻とトークン値を元に、次のパケットのスケジューリング予定時刻を計算し、キャッシュ部からの通知によりパイプライン処理部に同一のフローに属するパケットが存在する時は、それらのすべてのパケットをつなげた大パケットが仮想的に入力したものと見なして、次のパケットのスケジューリング予定時刻を計算することにより、同じフローに属するパケットが連続して入力しても、常に正しいスケジューリング時間を計算できるという効果がある。

【 0 0 8 5 】

また、本発明によれば、パイプライン処理部のパイプラインの段数（図 1 におけるパイプライン処理部内の処理ブロックの数）がいくつであっても実現可能で

あるので、伝送路の高速化にあわせてパイプラインの段数を増加させることで、将来の伝送路速度の高速化に際しても、任意のスケジューリングパラメータによるスケジューリング判定処理を実現することができる。

【 0 0 8 6 】

更にはまた、本発明による技術を用いずに高速な伝送路におけるシェーピング処理を実現する場合は、個々の高速なスケジューリング判定回路をフロー数だけ用意せざる得ないが、一般にフロー数は数  $k$  ～数  $10k$  にわたるため、極めて大規模な回路が必要であったのに比べ、本発明では、フロー単位に管理情報をメモリから読み出して行うパイプライン処理と、パイプライン処理の段数のみで規定されるキャッシュ部で実現されるため、小規模な回路で実現でき、結果として、ハードウェアコストの削減、消費電力の削減を実現することができるという効果がある。

【図面の簡単な説明】

【図 1】

本発明の実施例のブロック図である。

【図 2】

本発明によるシェーピング機能のシステム上の配置位置を示す図である。

【図 3】

基幹装置の構成とシェーピング判定部の位置関係を示す図である。

【図 4】

図 1 のブロック間信号の内容を示す図である。

【図 5】

(A) は管理メモリのフィールド構成を示し、(B) は内部レジスタのフィールド構成を示し、(C) はスケジューリング予定時刻メモリのフィールド構成を示す図である。

【図 6】

パイプライン処理部およびキャッシュ部の処理フローである。

【図 7】

図 6 の処理フローにおける変数 (パラメータ) を説明する図である。

【図 8】

本発明の実施例の動作を示すタイミングの一例である。

【図 9】

本発明の実施例の動作を示すタイミングの他の例である。

【図 1 0】

パイプライン処理を説明する図である。

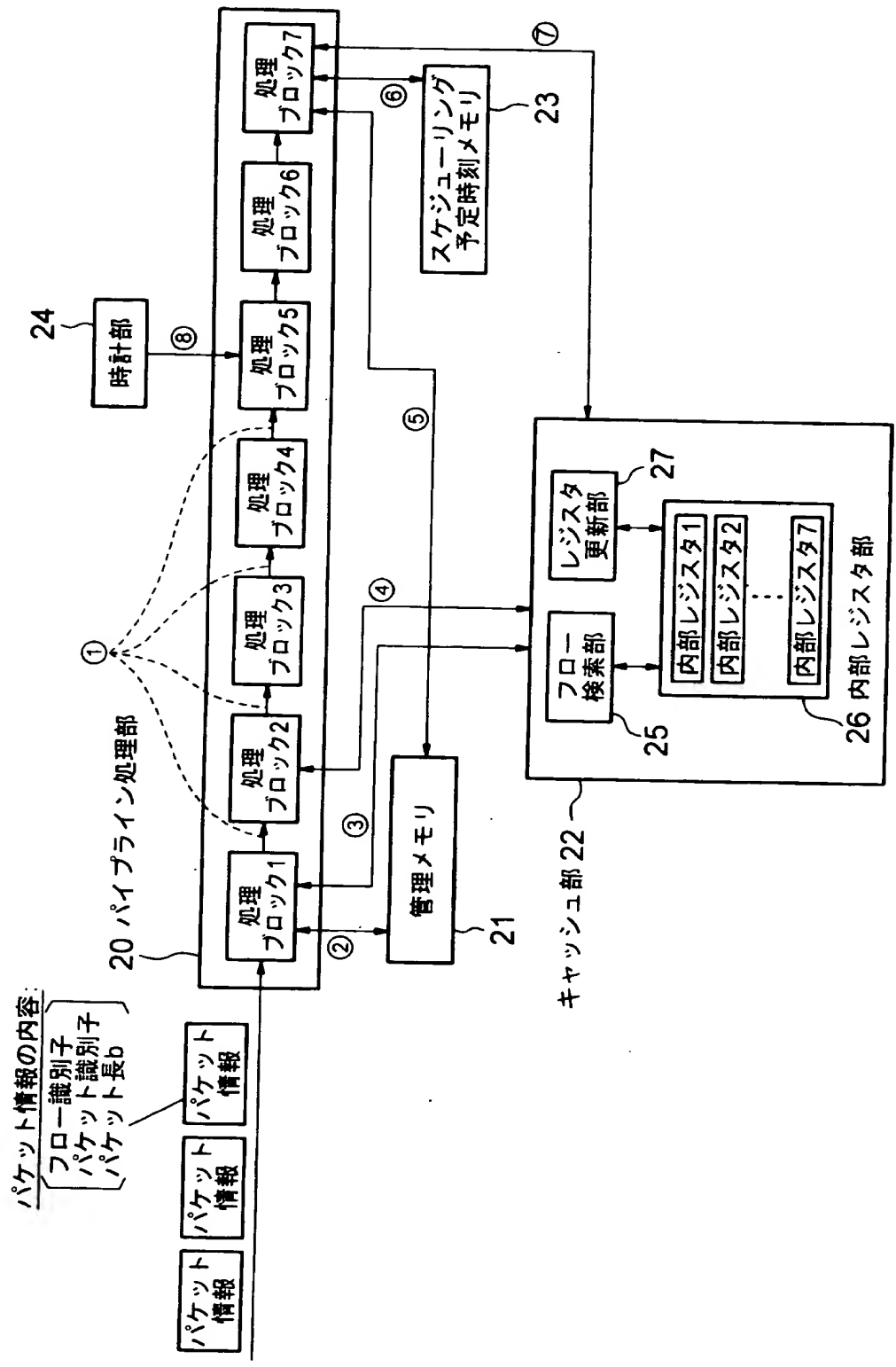
【図 1 1】

シェーピングに関するパイプライン処理の問題点を説明する図である。

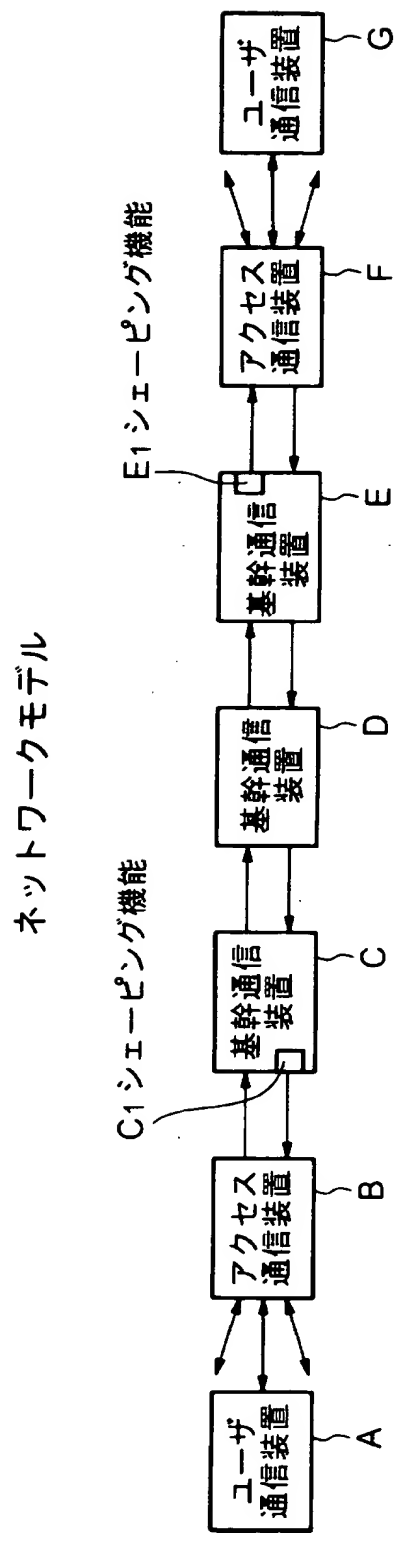
【符号の説明】

- 1 0 スイッチ部
- 1 2 バッファ部
- 1 3 シェーピング判定部
- 1 4 パケット読出し制御部
- 1 5 時計
- 2 0 パイプライン処理部
- 2 1 管理メモリ
- 2 2 キャッシュ部
- 2 3 スケジューリング予定時刻メモリ
- 2 4 時計部
- 2 5 フロー検索部
- 2 6 内部レジスタ部
- 2 7 レジスタ更新部

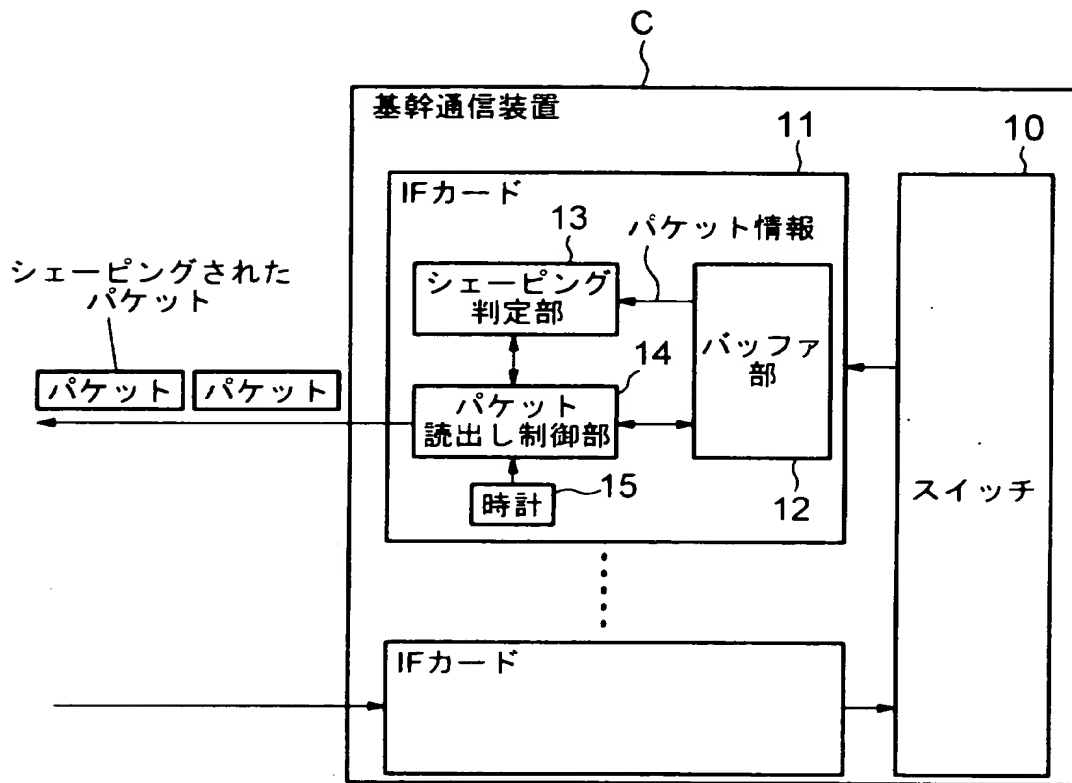
【書類名】 図面  
【図 1】



【図 2】



【図 3】





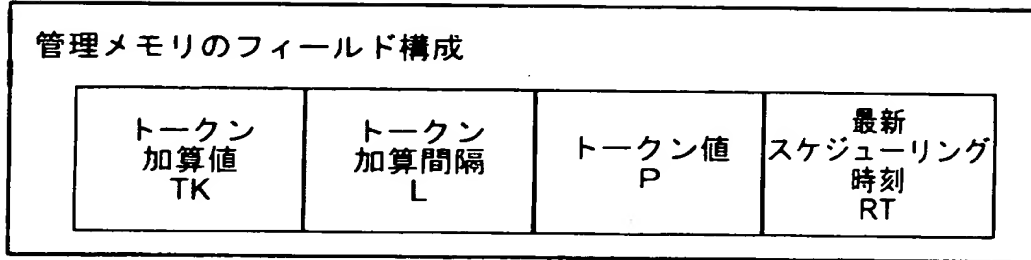
【図 4】

ブロック間信号の内容:

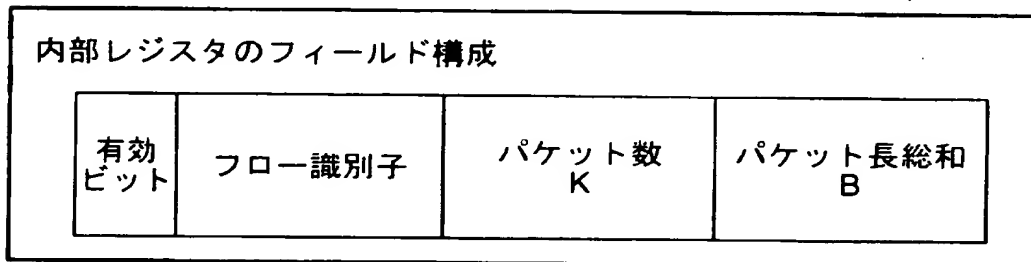
- ① パイプライン処理部内の処理ブロック間の信号  
フロー識別子、パケット識別子、TK、L、P、RT, 内部変換として f bit、X、Y、Z、W、f 2bit、Y2,
- ② 管理メモリのアドレスとしてフロー識別子、データとして TK、L、P、RT,
- ③ 処理ブロック1 → キャッシュ部にフロー識別子、キャッシュ部 → 処理ブロック1に K、B,
- ④ 処理ブロック2 → キャッシュ部にフロー識別子、パケット長b,
- ⑤ 管理メモリのアドレスとしてフロー識別子、データとしてトークン加算値TK、トークン加算間隔L、  
新たなトークン値として P=W、新たなスケジューリング予定時刻として RT=Z,
- ⑥ スケジューリング予定時刻Z、パケット識別子,
- ⑦ 処理ブロック7 → キャッシュ部にフロー識別子、パケット長b,
- ⑧ 現在時刻NT

【図 5】

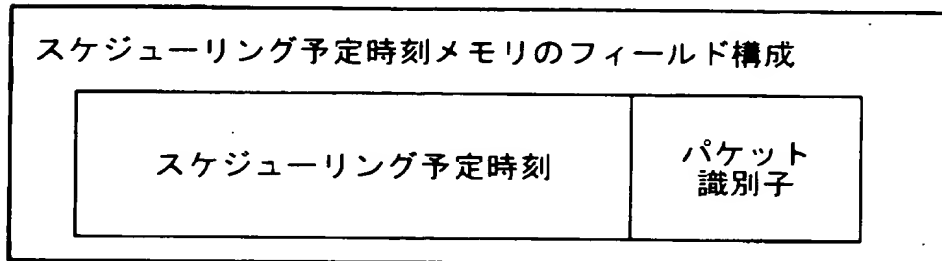
( A )



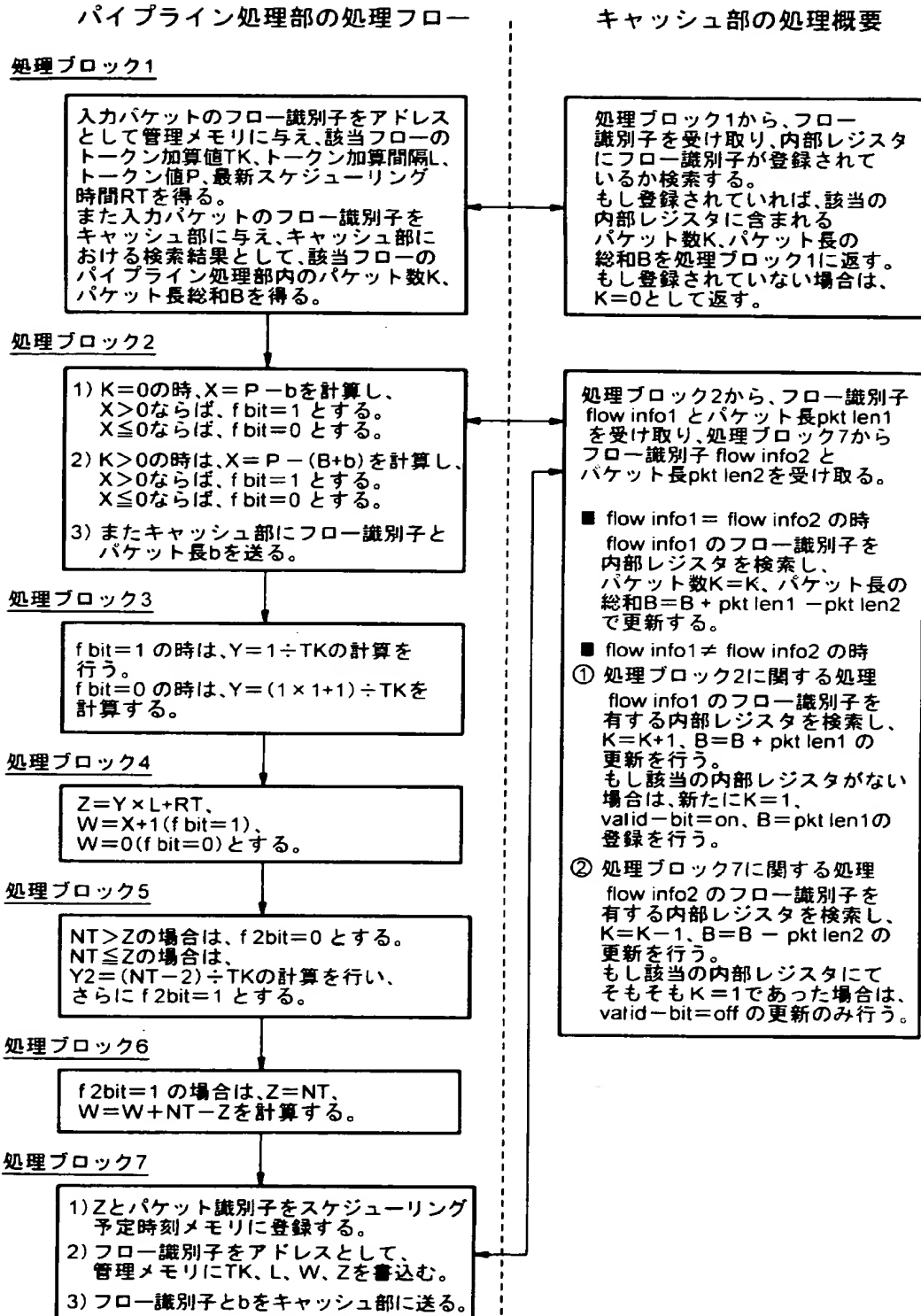
( B )



( C )



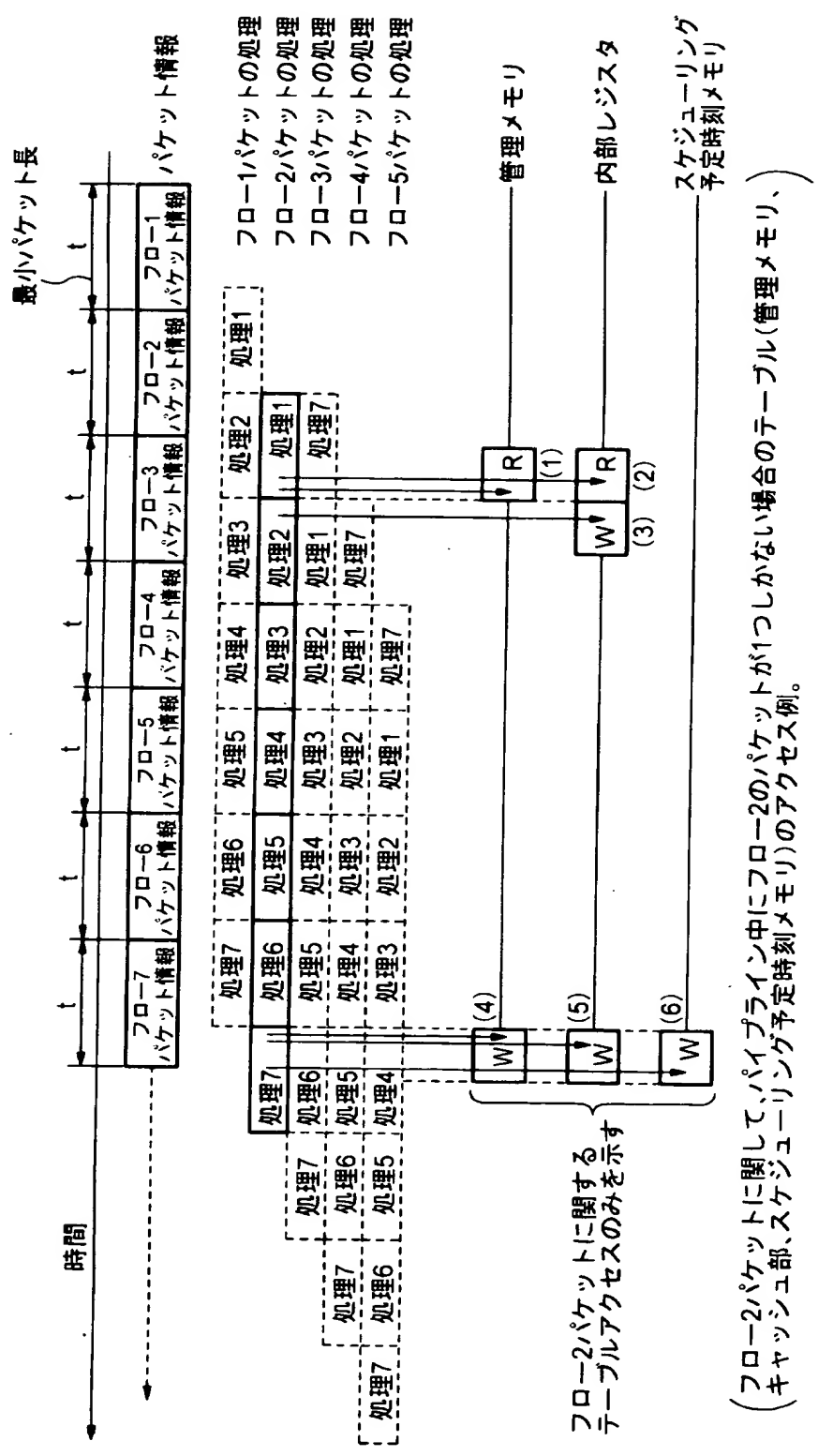
【図 6】



【図 7】

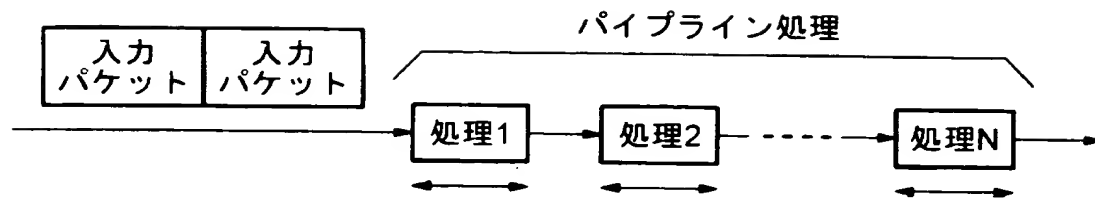
変数の説明:
<p>① 管理メモリからの読出し  TK=トークン加算値  L=トークン加算間隔  P=トークン値  RT=最新スケジューリング時間</p>
<p>② キャッシュ部からの読出し  K=パケット数  B=パケット長総和  valid-bit = レジスタ有効無効</p>
<p>③ 内部変数  f bit = トークンが足りているか否かの識別  X=足りないトークン量  Y=必要なトークン加算間隔の数  Z=新たなスケジューリング予定時刻  W=新たなトークン値  f 2bit = 新たなスケジューリング予定時刻  Zと現時刻の前後関係を示す。  Y2=Zと現時刻までに加算されるトークン  加算の回数。</p>
<p>④ その他  b=処理パケットのパケット長  NT=現在時刻</p>

【図 8】

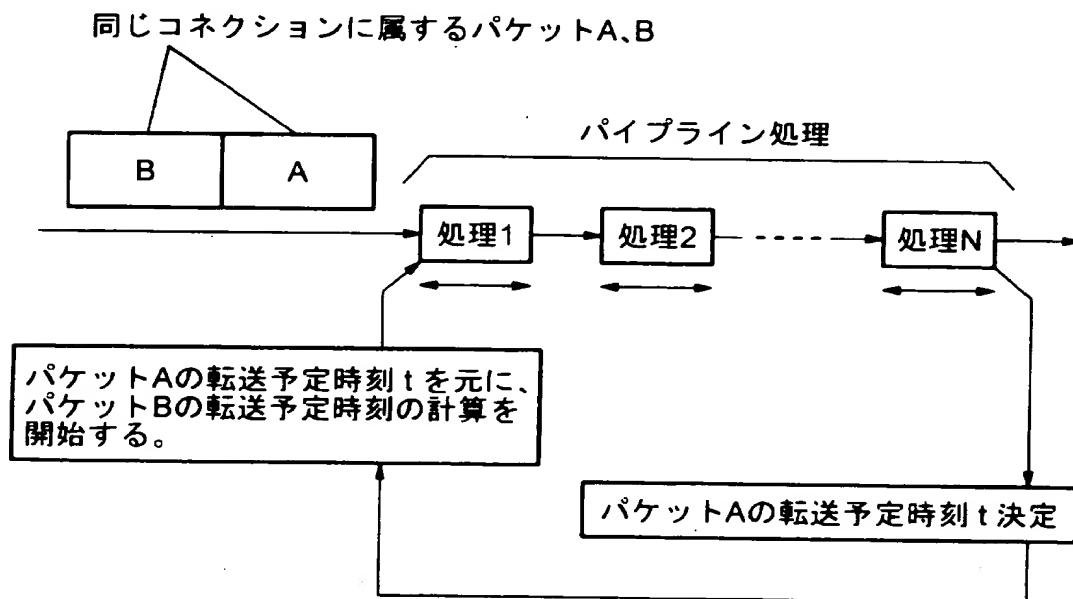




【図 1 0】



【図 1 1】



【書類名】 要約書

【要約】

【課題】 任意の速度のコネクションに対しても厳密なシェーピング処理を、簡易な回路構成の追加により実現可能としたパイプライン処理型シェーピング方式を得る。

【解決手段】 パイプライン処理部 2 0 の処理と連動するキャッシュ部 2 2 を用意し、このキャッシュ部 2 2 において、パイプライン処理部 2 0 で仕掛かり中のパケットのフロー情報を管理し、同じフローに属するパケットがある場合は、キャッシュ部 2 2 が該当のパケットを全てつなぎ合わせた仮想的なパケットを想定したパラメータをパイプライン処理部 2 0 に渡し、パイプライン処理部 2 0 ではこの仮想的なパラメータを元にパイプライン処理を実行することにより、任意のピークレート（同じフローに属する入力パケット間隔の逆数）の速度を有するフローに対しても、またどのような高速な伝送路インタフェースにおいても、常にシェーピングによるスケジューリング予定時刻をリアルタイムで計算できる。

【選択図】 図 1



出 願 人 履 歴 情 報

識別番号 [000004237]

1. 変更年月日 1990年 8月29日

[変更理由] 新規登録

住 所 東京都港区芝五丁目7番1号

氏 名 日本電気株式会社